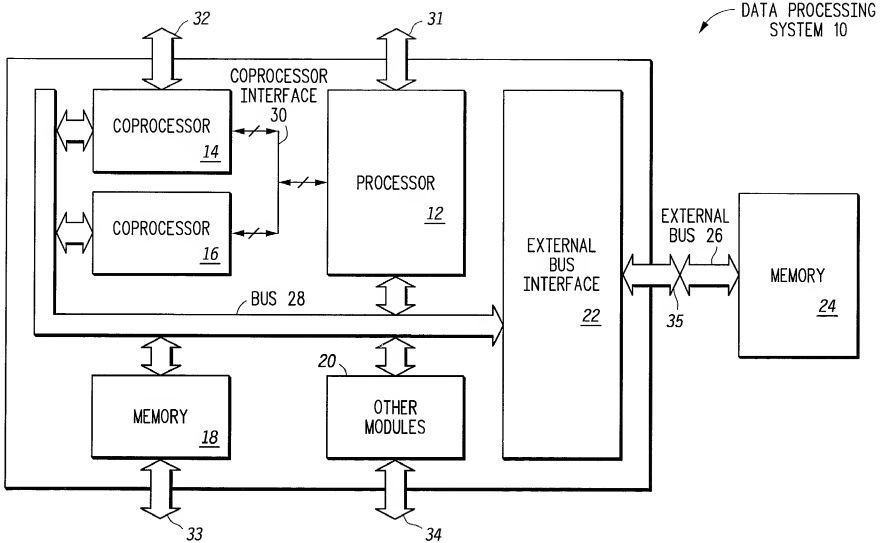

FIG.1

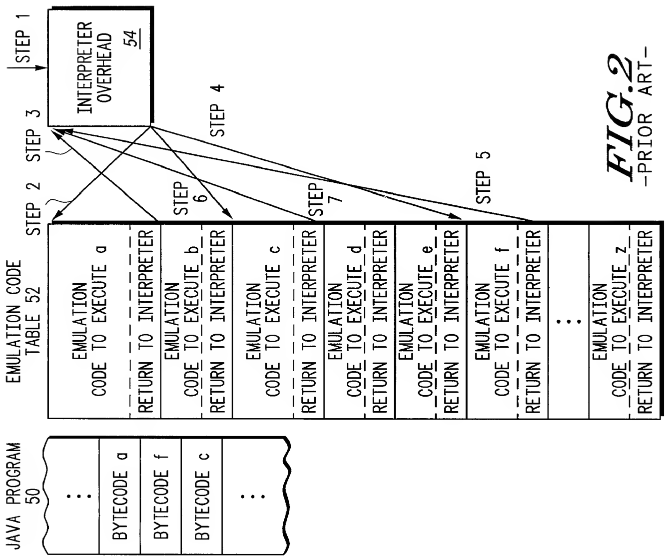

FIG.2
-PRIOR ART-

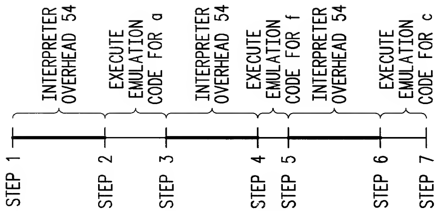

FIG.3
-PRIOR ART-

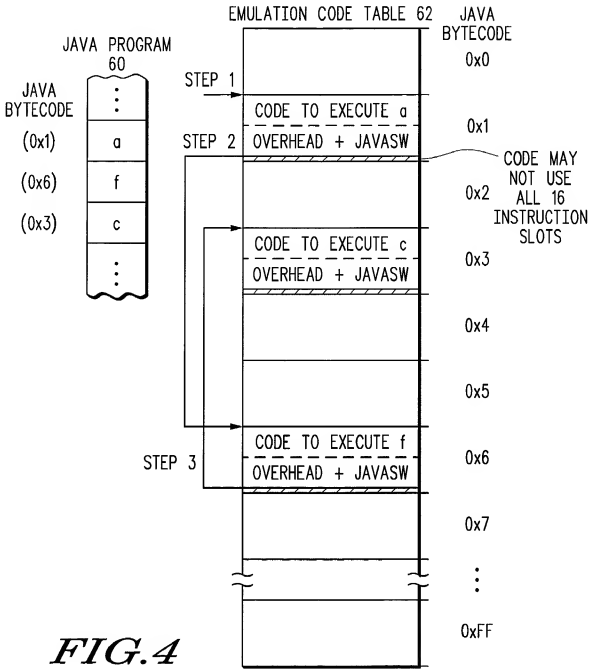
JAVA PROGRAM 60
JAVA BYTECODE
(0x1)
(0x6)
(0x3)
a
f
c
EMULATION CODE TABLE 62
JAVA BYTECODE
STEP 1
STEP 2
STEP 3
CODE TO EXECUTE a
OVERHEAD + JAVASW
CODE TO EXECUTE c
OVERHEAD + JAVASW
CODE TO EXECUTE f
OVERHEAD + JAVASW
CODE MAY NOT USE ALL 16 INSTRUCTION SLOTS
0x0
0x1
0x2
0x3
0x4
0x5
0x6
0x7
0xFF

FIG.4

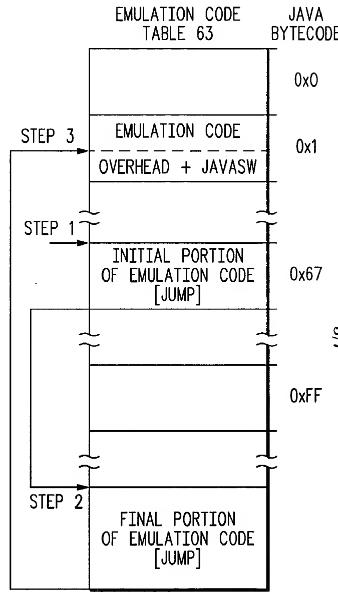
EMULATION CODE TABLE 63
JAVA BYTECODE
EMULATION CODE
OVERHEAD + JAVASW
INITIAL PORTION OF EMULATION CODE [JUMP]
FINAL PORTION OF EMULATION CODE [JUMP]
STEP 1
STEP 2
STEP 3
0x0
0x1
0x67
0xFF

FIG.5

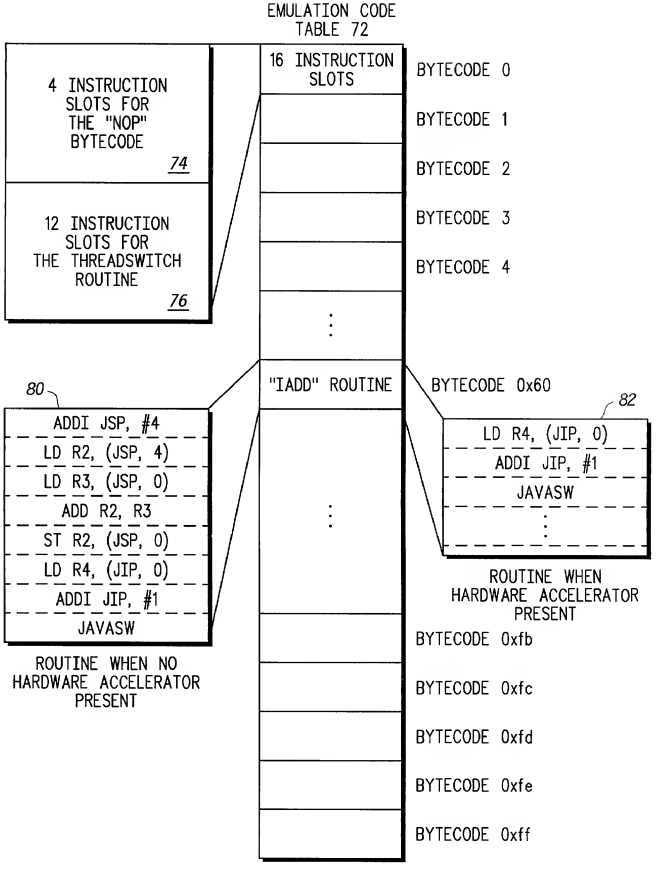
EMULATION CODE TABLE 72
4 INSTRUCTION SLOTS FOR THE "NOP" BYTECODE
74
12 INSTRUCTION SLOTS FOR THE THREADSWITCH ROUTINE
76
16 INSTRUCTION SLOTS
BYTECODE 0
BYTECODE 1
BYTECODE 2
BYTECODE 3
BYTECODE 4
"IADD" ROUTINE
BYTECODE 0x60
BYTECODE 0xfb
BYTECODE 0xfc
BYTECODE 0xfd
BYTECODE 0xfe
BYTECODE 0xff
80
ADDI JSP, #4
LD R2, (JSP, 4)
LD R3, (JSP, 0)
ADD R2, R3
ST R2, (JSP, 0)
LD R4, (JIP, 0)
ADDI JIP, #1
JAVASW
ROUTINE WHEN NO HARDWARE ACCELERATOR PRESENT
82
LD R4, (JIP, 0)
ADDI JIP, #1
JAVASW
ROUTINE WHEN HARDWARE ACCELERATOR PRESENT

FIG.6

JAVASW JAVA INTERPRETER SWITCH

OPERATION: JUMP TO BYTECODE EMULATION SEQUENCE;

```
91 ~ IF (SWCOUNT!<0)
92 ~ PC<-[((PC+2) & 0xffffe000) | (R4<<5)];
93 ~ ELSE
94 ~ PC<-[((PC+2) & 0xffffe000) | (0x8)];
95 ~ SWCOUNT--;
```

ASSEMBLER
SYNTAX: JAVASW RX

DESCRIPTION: JUMP TO BYTECODE EMULATION SEQUENCE. THE LOW ORDER 13 BITS OF THE VALUE OF PC+2 ARE FORCED TO ZERO, AND BASED ON THE STATE OF THE SWCOUNT (IN REGISTER R12), EITHER A SCALED VALUE IN REGISTER RX IS LOGICALLY "OR'ED", OR A CONSTANT VALUE 0x8 OR'ED, AND INSTRUCTION EXECUTION RESUMES AT THE NEW PC VALUE. THE SWCOUNT REGISTER R12 IS DECREMENTED. NOTE THAT BECAUSE PC+2 IS USED AS THE BASE VALUE, A JAVASW INSTRUCTION SHOULD NOT BE USED IN THE LAST INSTRUCTION OF INSTRUCTION GROUP 255.

CONDITION CODE: INAFFECTED

INSTRUCTION FORMAT:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0

90

INSTRUCTION FIELDS:

JAVA INTERPRETER SWITCH SUPPORT IS PROVIDED BY THE JAVASW INSTRUCTION. THIS INSTRUCTION CAUSES CONTROL FLOW TO BE DIRECTED INTO A TABLE OF INSTRUCTION GROUPS. MOST GROUPS CONSISTS OF 16 INSTRUCTIONS AND CORRESPOND TO A SINGLE JAVA BYTECODE.

FIG. 7

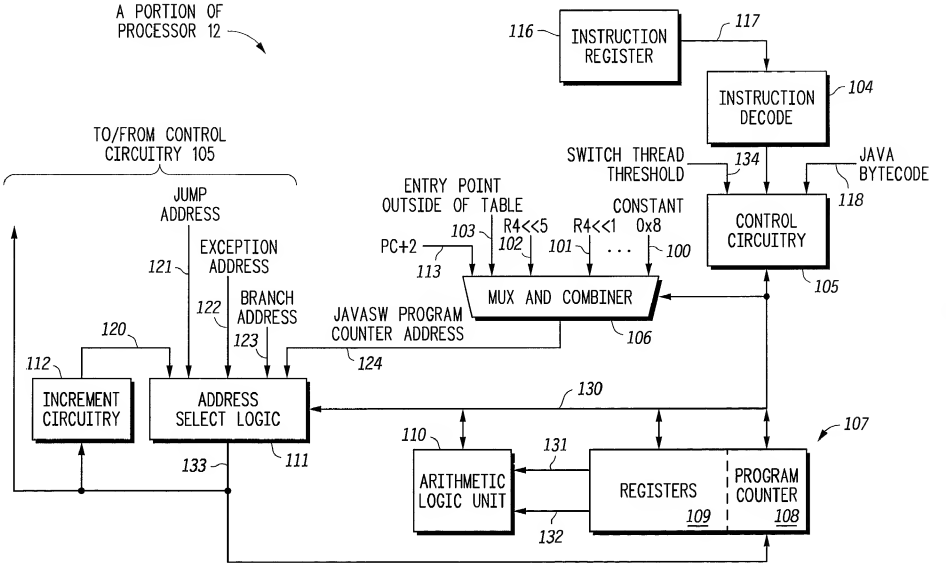

FIG.8

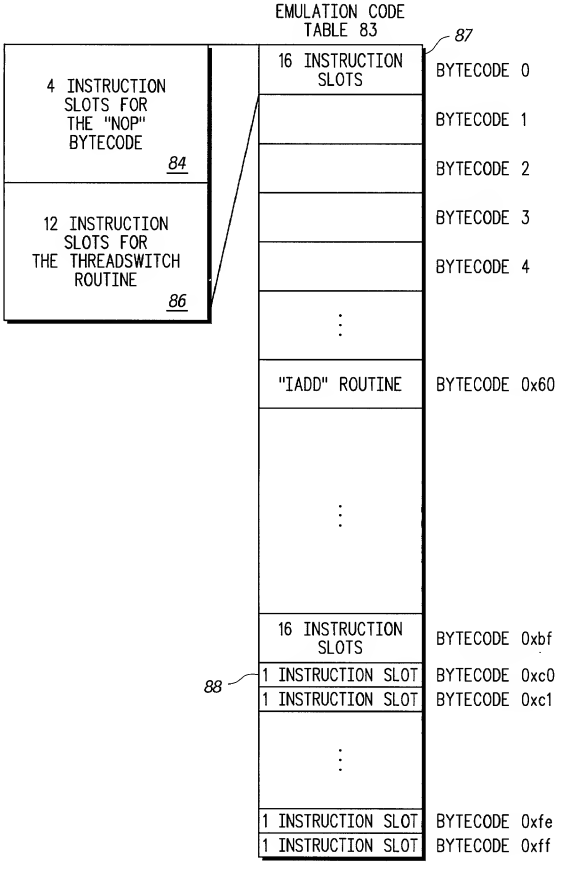
EMULATION CODE TABLE 83
87
4 INSTRUCTION SLOTS FOR THE "NOP" BYTECODE
84
12 INSTRUCTION SLOTS FOR THE THREADSWITCH ROUTINE
86
16 INSTRUCTION SLOTS
BYTECODE 0
BYTECODE 1
BYTECODE 2
BYTECODE 3
BYTECODE 4
"IADD" ROUTINE
BYTECODE 0x60
16 INSTRUCTION SLOTS
BYTECODE 0xbf
88
1 INSTRUCTION SLOT
BYTECODE 0xc0
1 INSTRUCTION SLOT
BYTECODE 0xc1
1 INSTRUCTION SLOT
BYTECODE 0xfe
1 INSTRUCTION SLOT
BYTECODE 0xff

FIG. 9